

2018년 1학기

창의적설계

최종보고서

제목: 시각장애인을 위한 점자 인식기

제출일: 2018년 6월 7일

지도교수명	홍상훈 교수님
조명	8조
조원명	김남혁, 백지현, 진장혁
대표연락처	E-MAIL: nh9719@naver.com 핸드폰: 010-7150-9719

1. 설계 작품 배경

A. 설계 작품의 중요성/필요성 (200자 이내로 요약)

시각장애인에게 있어서 점자는 정보를 얻기 위해 반드시 필요한 삶의 도구이다. 그리고 주변에서 점자를 쉽게 찾아볼 수 있기 때문에 많은 사람들이 시각장애인의 점자 인식 정도가 높다고 생각한다. 하지만 실제로는 시각장애인의 90% 이상이 점자를 스스로 읽지 못하며 점자를 통한 정보의 인식 및 전달은 타인의 도움을 받아야만 가능하다.

본 설계 작품은 점자를 스스로 읽지 못하는 시각장애인의 불편함을 해소하는 것이 목적이다. 점자를 인식하여 이에 맞는 글자를 음성으로 알려주기 때문에 타인의 도움 없이도 점자를 통한 정보 인식이 가능하다.

B. 설계 작품의 창의성 (200자 이내로 요약)

시각장애인이 사용하고 있는 점자는 3X2 형태의 6개 점으로 구성되며 한글의 초성, 종성, 모음을 표현하기 위해 각기 다른 형태로 돌출되어 있다. 따라서 한 글자를 나타내는 점자를 인식하기 위해서는

6개의 점들이 어떻게 구성되어 있는지를 알아야 했다.

본 설계 작품은 각 점들이 돌출된 정도를 정확히 인식하여 신호를 전달해야 되기 때문에 정밀한 제작이 가능한 3D 프린팅 기술을 활용했다. 2열로 이루어진 점자를 한 번에 인식하는 것 보다 1열씩 인식하는 것이 효율적이라고 생각하여 3개의 홈을 판 인식기를 제작했다. 인식기 위에는 적당한 높이로 PCB 기판을 올려 놓고 점자를 지나가면 기판에 접촉되어 물리적인 신호를 전기적인 신호로 바꿔줄 수 있도록 설계했다.

2. 설계 작품에 대한 문제 정의

- (1) 실제 점자와 동일한 크기의 점을 인식할 수 있는가?
- (2) 인식기의 홈이 점의 중앙과 정확하게 일치하는가?
- (3) 점자를 지나갈 때 미세한 차이로 인한 인식 오류는 없는가?
- (4) 인식을 통해 점자에 대한 글자를 정확히 음성으로 출력할 수 있는가?

3. 설계 작품 내용

[a] 시작품에 대한 아이디어 창출

1) 번역기 – 평소 읽을 수 없는 언어를 번역하여 알려주고 이를 음성으로 들을 수도 있다.

2) 압력 센서 – 물리적 신호를 전기적 신호로 바꿔 sensing한다.

3) 2진법 - 0과 1로 표기하는 방법.

전기적으로 보면 open인 경우 0, short인 경우 1로 생각한다.

[b] 시작품에 대한 아이디어 다듬기

1) 3D 프린터를 이용하여 3개의 홈을 가진 인식기 틀 제작

2) 점자에서 점이 있는 곳은 1, 없는 곳은 0으로 지정

3) 점이 있는 곳은 PCB 기판과 맞닿으면 Short 될 수 있도록 설계

4) 아두이노로 입력 받은 신호를 이용하여 음성으로 출력

[c] 시작품에 대한 아이디어 판정

1) 점자를 지나가며 인식해야 하므로 3개씩 차례로 인식하도록 인식기 제작

2) 6개의 점자가 인식되면 0과 1로 구성된 2진수로 입력을 받고 10진

수로 변환하여 해당 글자가 무엇인지 확인

- 3) 하나의 점도 돌출되지 않은 열이 있기 때문에 다른 글자로 잘못 인식할 수 있지만 적절한 코딩을 통해 오류 방지
- 4) 적절한 코딩을 통해 미세한 차이로 인한 인식 오류가 발생하지 않도록 설계

[d] 시작품 목표 사양 및 설계 방법

1) 설계 방법

- a. 3D 모델링 소프트웨어를 활용하여 인식기와 점자 디자인
- b. 3D 프린터를 활용하여 모델 출력
- c. PCB 기판에 점자 인식 시 신호를 전달해 줄 수 있는 회로 제작
- d. 코딩을 통해 인식된 점자 조합 및 음성 출력

2) 목표 사양

- a. 정확한 점자 인식
- b. 인식된 점자를 통해 정확한 단어로 조합
- c. 점자와 인식기 크기 최소화

4. 설계 요구사항 정의

1) 성능적 제한 요구사항

긴 점자도 빠르고 정확하게 인식할 수 있도록 프로그래밍 알고리즘을 설계한다. 점자를 지나가면서 점이 없는 곳은 0, 있는 곳은 1로 인식하도록 한다. 하나의 글자는 0 또는 1로 표현된 여섯 자리 이진수로 받아들이고 이를 십진수로 변환하여 해당 숫자가 나타내는 글자가 무엇인지 확인할 수 있도록 한다. 글자를 정확하게 읽어낸 후 단어로 조합하고 오디오 모듈에서 이에 해당하는 음성 파일을 재생시켜 스피커로 출력될 수 있도록 설계한다.

데이터를 입력 받은 후 글자로 조합하여 음성으로 출력하는 데까지 걸리는 delay 시간은 5초 이내로 하고, 정확하게 점자를 인식하기 위한 센서의 error rate은 3% 미만으로 한다.

2) 비용적 제한 요구사항

Arduino Mega, Audio-Sound Breakout(WTV020SD), 아두이노 호환 디지털 스피커 모듈 등을 포함하여 필요한 부품들을 제공되는 지원금 20만원 내에서 구입하여야 한다.

3) 환경(동작온도 등)적 제한 요구사항

어느 정도 빠르거나 느린 속도로 sliding하여도 점자를 정확하게 인식 가능해야 된다. 하나의 열이 모두 비어있는 경우를 구분하기 위해 일정한 속도로 sliding 해야 된다는 제한사항이 있다. 그리고 야외에서도 이용해야 될 경우가 있는 점자 인식기 특성상 습도나 온도에 영향을 받지 않고 일정한 동작이 가능해야 한다. 습도 기준은 0%에서 비 오는 날 평균 습도인 90%로 설정하고, 온도는 -10도에서 40도로 설정한다.

4) 디자인적 제한 요구사항

- 점자는 점자 표준 규격으로 디자인하면 너무 작아서 필요한 부품을 구하는 데 어려움이 있고 점자를 인식하는 과정에서 오차가 커질 것을 고려하여 표준보다 큰 크기로 제작한다.
- 항상 소지하고 다니기에 부담되지 않는 무게와 크기로 제작한다. 단, 크기가 너무 작으면 사용자가 잡고 sliding하기 불편할 수 있기 때문에 손의 크기를 고려하여 적절한 인식기 크기를 결정한다.
- sliding 하는 과정에서 점자를 지나가는 길을 벗어나는 일이 없도록 인식기

에 점자 고정 틀을 디자인한다.

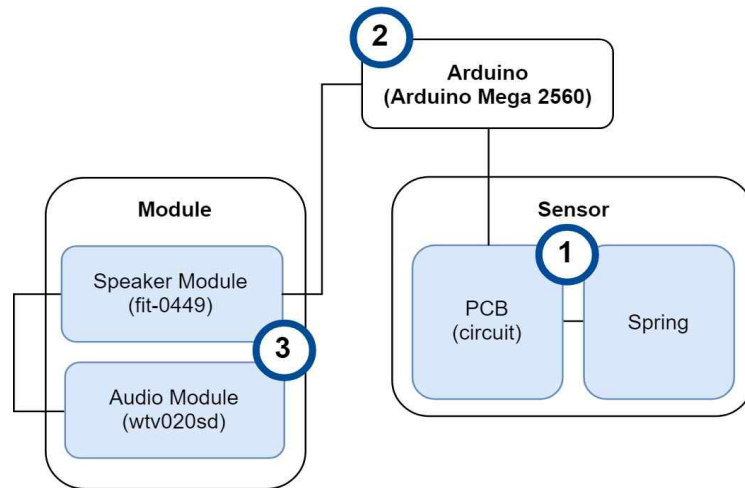
- 점자 높이에 맞게 스프링이 올라가서 PCB 기판에 정확하게 접촉될 수 있도록 디자인 한다.

5. 설계 작품 요약

- 1) 즉시 점자를 인식하여 음성으로 출력하는 실시간 점자 인식기
- 2) 부드러운 sliding 방식으로 쉽고 빠르게 사용 가능한 점자 인식기
- 3) 점자의 재질이나 주변 환경에 상관없이 사용 가능한 점자 인식기
- 4) 작은 크기로 휴대하기 편리한 휴대용 점자 인식기

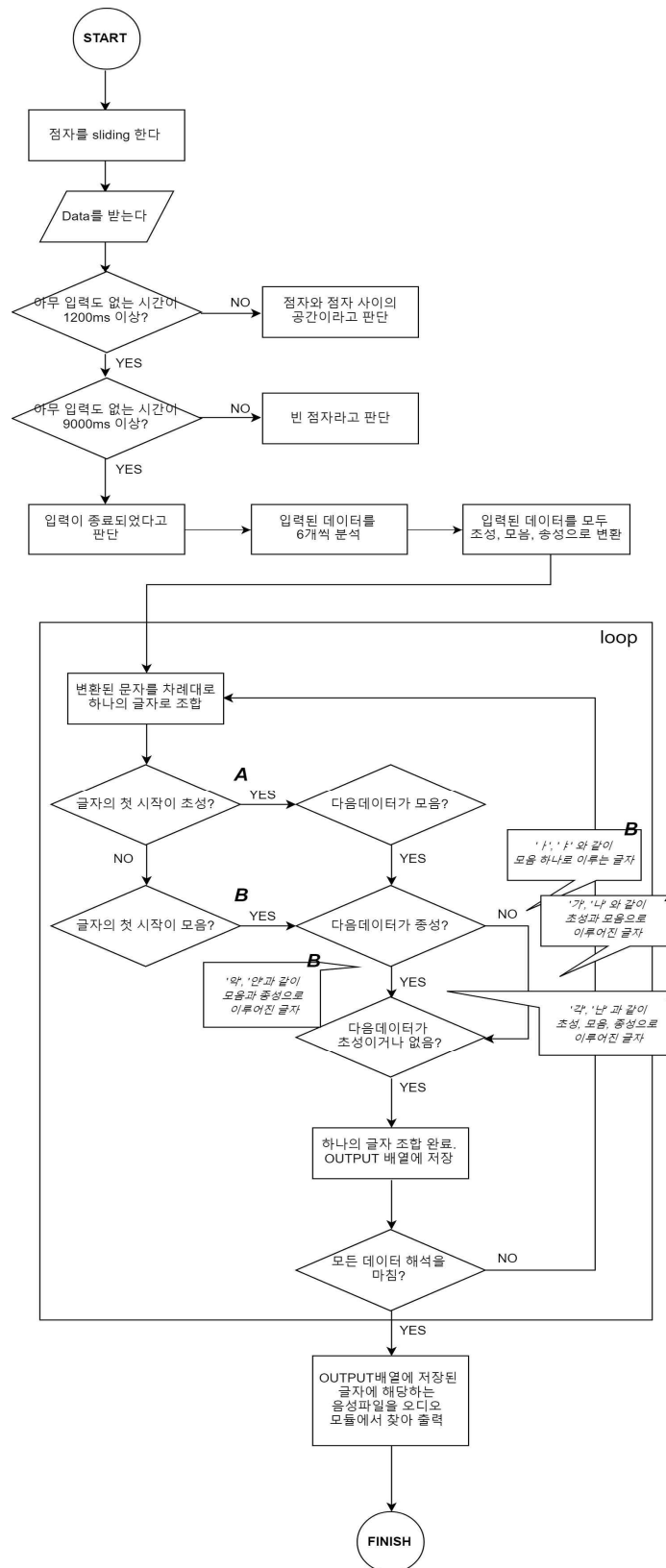
6. 시스템 설계 규격

1) Block Diagram

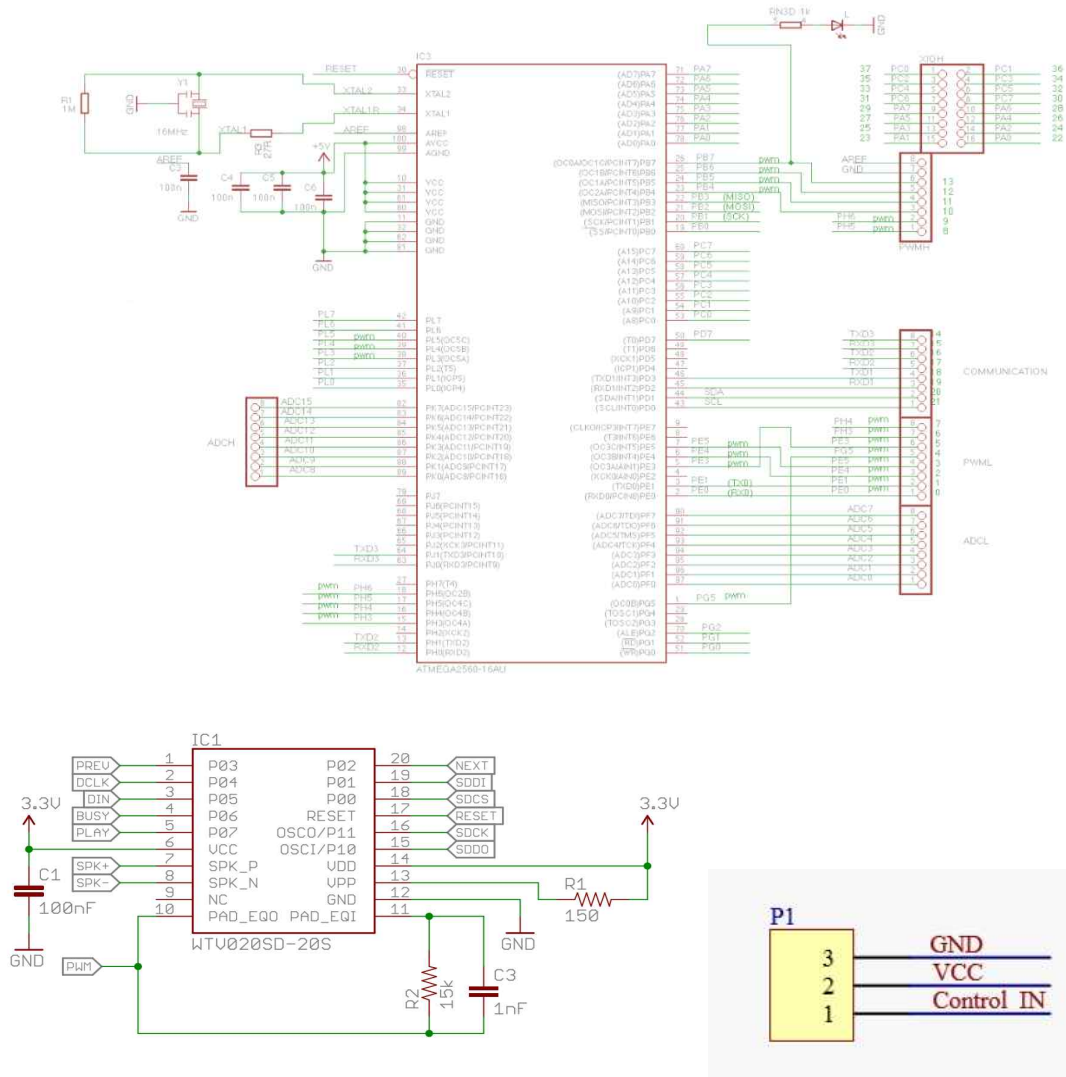


- ① 2mm 높이로 제작된 점을 지나면 스프링이 2mm만큼 올라와 PCB 기판에 정확히 닿아서 short되며 신호가 전달된다. 스프링이 닿지 않는 부분은 0V, 닿는 부분은 3.3V의 디지털 데이터를 갖는다.
- ② 입력된 데이터를 아두이노를 사용하여 처리한다. 데이터는 0과 1을 사용하여 6개씩 저장한 뒤 이를 10진수로 변환하여 다시 배열에 저장한다. 입력을 마치면 앞에서부터 차례대로 10진수 데이터를 읽어와 미리 저장되어있는 49개의 점자 데이터를 비교하여 어떤 문자인지 판단한다. 판단한 문자들을 글자로 조합한 뒤 배열에 저장한다.
- ③ 모든 조합을 마치면, 배열에 저장되어있는 글자의 녹음파일을 찾아 오디오와 스피커 모듈을 사용하여 차례대로 음성으로 출력한다.

2) Flow Chart



3) Full Hardware Schematic



7. 모듈 설계 규격

1) 오디오 모듈(WTV020SD)

오디오 모듈은 필요한 단어를 음성으로 저장해 놓기 위해 필요한

모듈이다. 해당 모듈에는 SD 카드 슬롯이 존재하므로 SD 카드에 원

하는 음성 파일을 로드하여 사용한다.

Feature

- 임베디드 오디오-사운드 어플리케이션을 위한 저비용 모듈
- 6-36kHz 샘플링 레이트의 4bit ADPCM 파일 포맷 지원
- 스피커의 직접 연결($8\Omega/0.5W$) -> 차동 라인 2 PWM 출력
- 전용 16비트 DAC/PWM 오디오 출력은 외부 증폭기로 사용
- 오디오, 음성 및 음악파일(최대 512파일) 저장을 위한 SD 메모리 카드 어댑터 존재

2) 스피커 모듈(FIT0449)

점자를 지나가면서 얻어진 데이터의 분석이 끝나면 입력된 데이터를 초성, 모음, 종성 중 해당 되는 것으로 변환하고 각각을 조합하여 하나의 글자를 완성한다. 이를 오디오 모듈에 저장되어 있는 음성 파일과 matching 시켜 인식된 문장의 글자를 한 글자씩 소리로 출력한다.

Feature

- Operating Voltage: 2.0-5.5V
- Interface Type: Digital

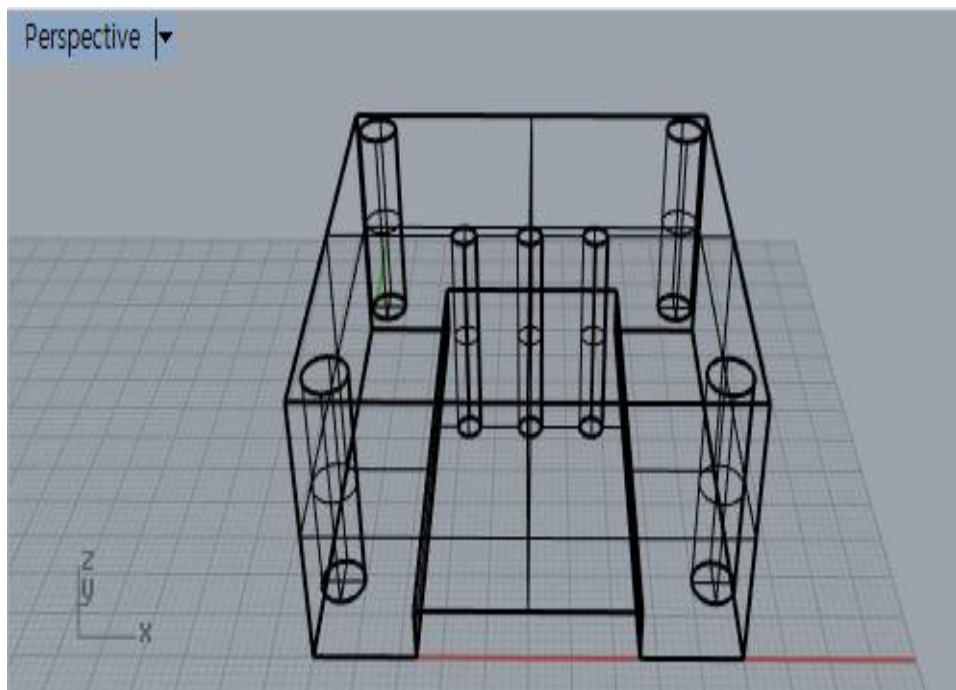
- Module Size: 40x40(mm)
- Support Gravity Interface

8. 상세설계 규격

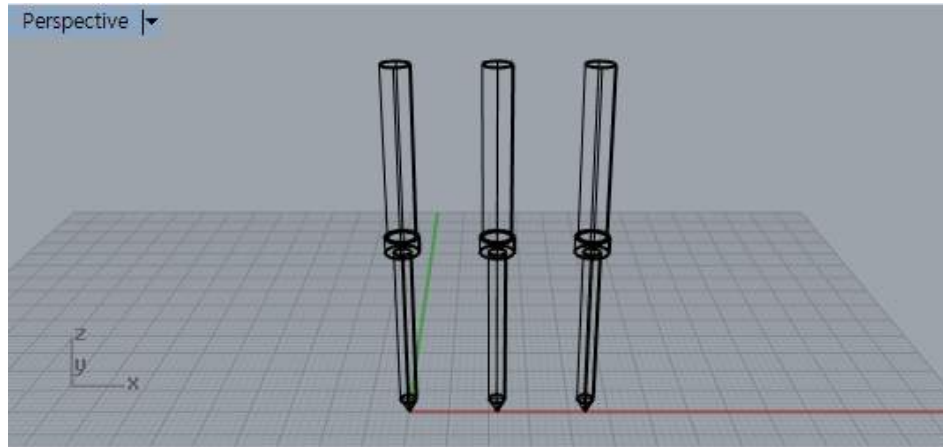
1) 점자 인식기

3D 프린팅 기술을 활용하여 아래의 규격으로 점자 인식기를 제작한다.

<점자 인식기 도면>

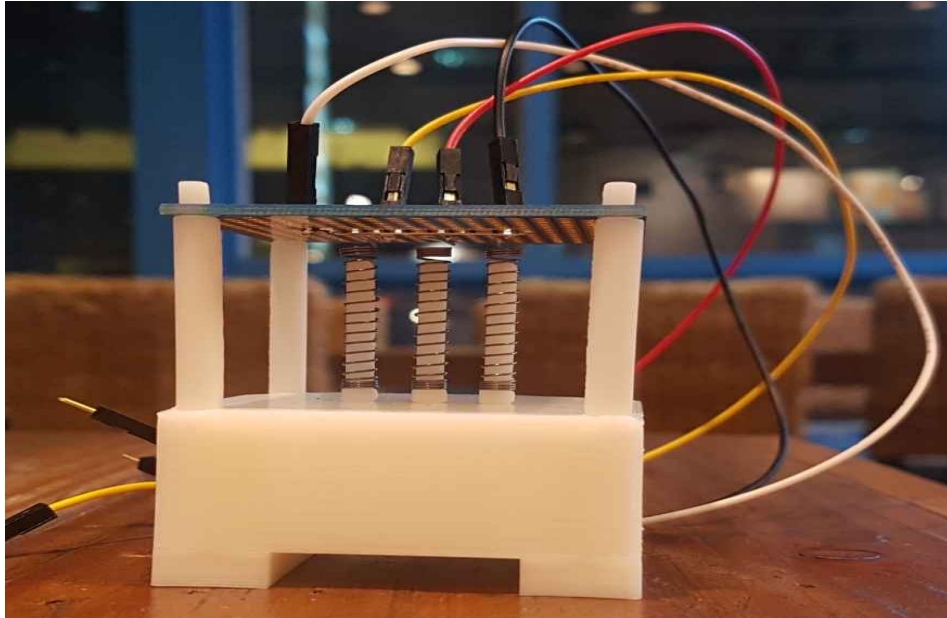


<점자를 지나갈 막대기 도면>



- 인식기 전체 규격: 50 x 50 x 25 (mm)
- 막대기가 들어가는 홈의 지름: 3(mm)
- 점자가 지나는 길: 26 x 50 x 5 (mm)
- 막대기 길이: 46(mm)
- 스프링 길이: 24(mm)
- 막대기가 올라가는 높이: 3mm(점자 판의 높이)+2mm(점의 높이)

<3D프린팅으로 완성된 점자 인식기>



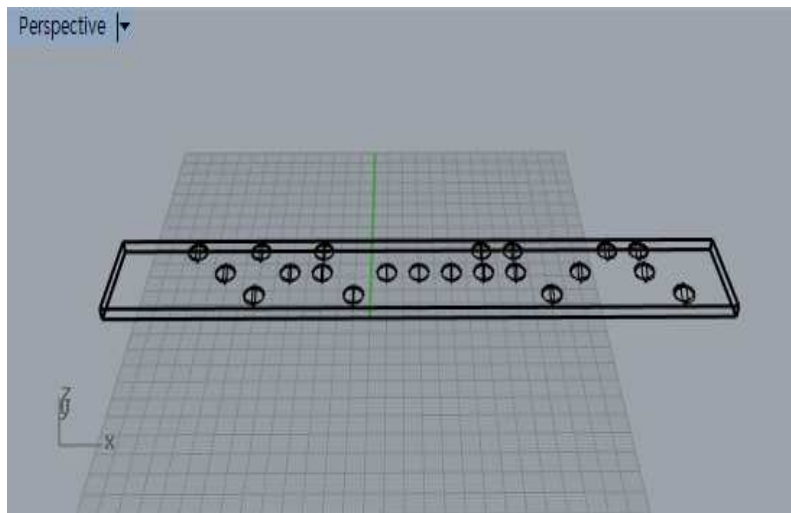
3D 프린팅 한 인식기 모델의 홈에 세 개의 막대기를 꽂고 인식기 위쪽으로 나온 막대기에 스프링을 끼워 같이 움직일 수 있도록 한다. 위와 같이 막대기를 꽂으면 인식기에는 2mm 높이의 스프링 받침대와 24mm 길이의 스프링이 세 개의 홈 위쪽으로 올라와 있는 상태이다.

점자를 지나갈 때 스프링이 추가적으로 올라가는 높이를 고려하여 PCB 기판을 인식기 지지대에 끼워 놓는다. 점이 있는 곳을 지나가면 스프링이 점자 높이만큼 올라가고 PCB 기판에 접촉 및 short 되어 데이터를 전송할 수 있도록 제작한다.

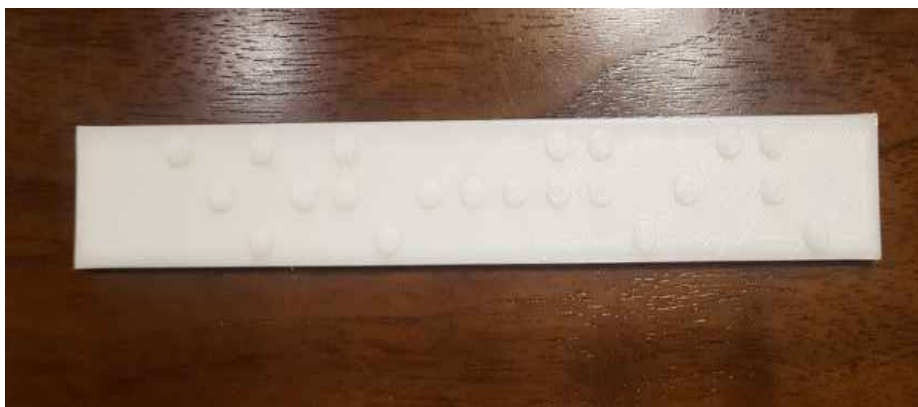
2) 점자

- 점자 한 개의 규격: 반 타원형 길이 6(mm), 높이 2(mm)
- 한 글자의 판 규격(점자 6개 기준): 25 x 20 x 3 (mm)

<점자 도면>



<점자 모형>

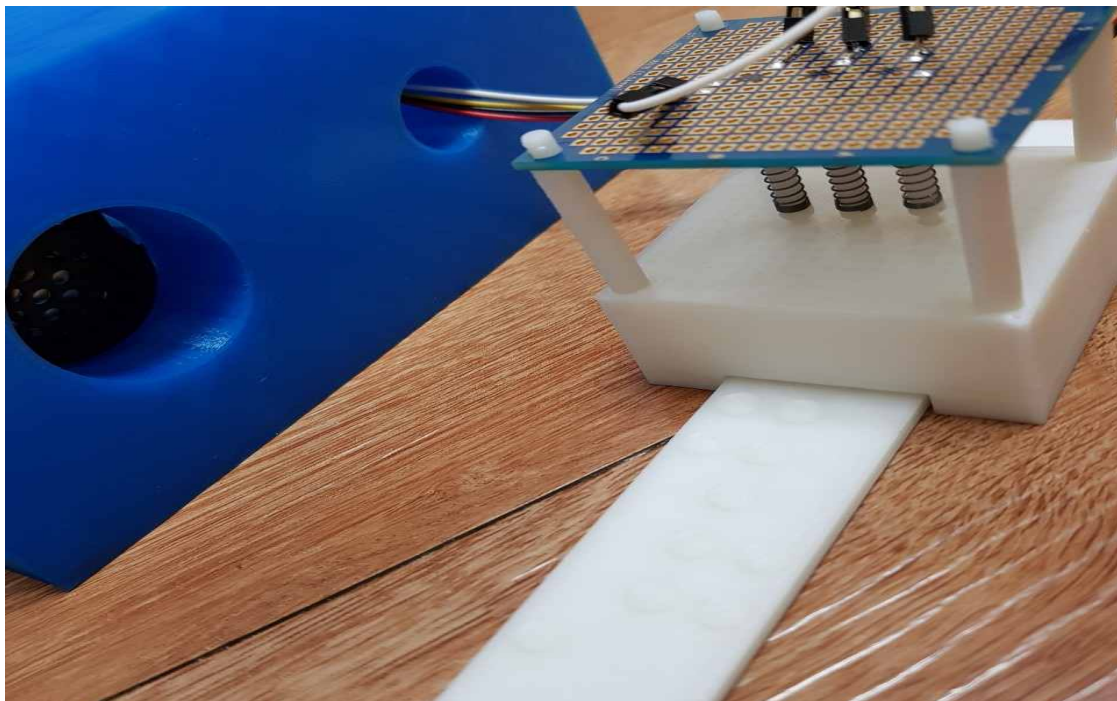
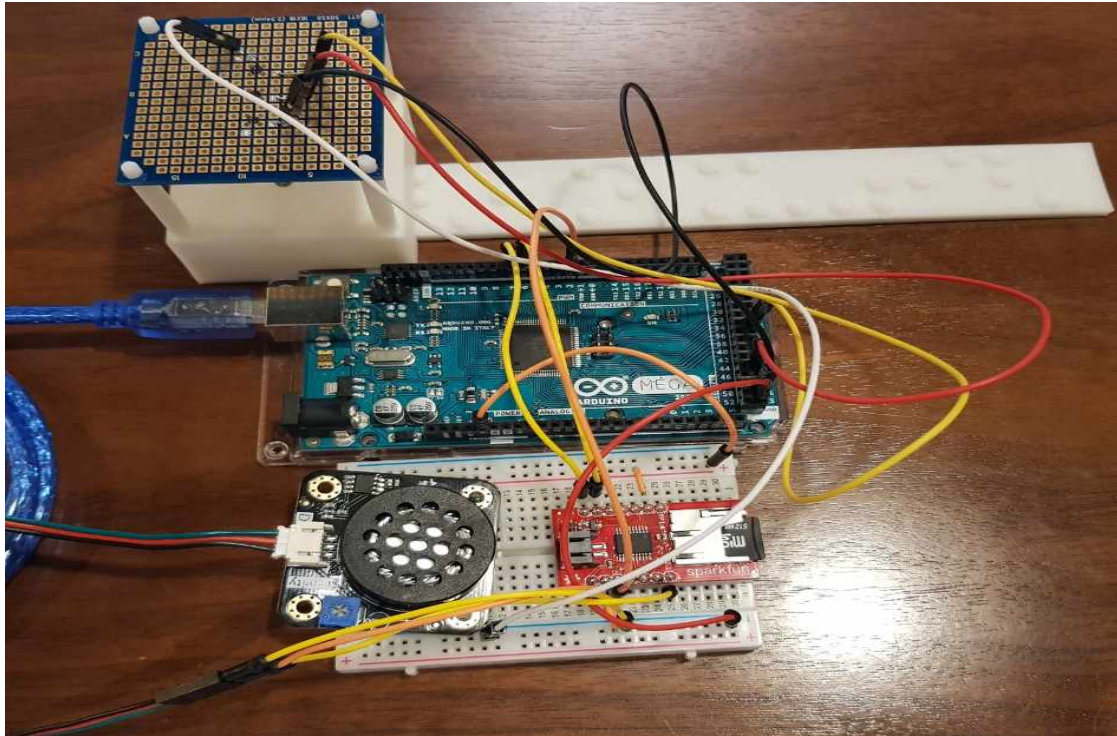


점자판의 앞 부분에 약 2cm의 빈 공간을 두어 인식기를 미리 올려놓을 수 있도록 디자인

3) 코드 개요

```
1  #include <FreeRTOS_AVR.h>
2
3  #define SignalPin1 53    // 검
4  #define SignalPin2 47    // 빨
5  #define SignalPin3 37    // 노
6
7  #define Blank_JudgeTime 1200    // 이상 입력이 들어오지 않아야 빈 점자로 판단
8  #define Nosalnal_EndTime 9000    // 이상 입력이 없으면 입력이 종료되었다고 판단
9  #define Debouncing_Time 300
10
11  const unsigned char Initial[64] = { ... }
12
13  const unsigned char Medial[64] = { ... }
14
15  const unsigned char Final[64] = { ... }
16
17  /* ... */
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45  int blank_counter = 0; // 빈 점자인지 점자와 점자 사이의 공간인지 판단하기 위한 counter
46  int nosignal_counter = 0; // 일정 시간 이상 입력이 들어오지 않으면 입력 종료를 알리기 위한 counter
47  int readIndex = 0; // inputBuffer에 저장된 데이터 처음부터 끝까지 순차적으로 읽는 인덱스
48  int speakIndex = 0; // speakBuffer에 저장된 데이터 처음부터 끝까지 순차적으로 읽는 인덱스
49
50  int inputBuffer[100] = { 0 }; // 점자로 해석할 데이터 저장을 위한
51  int speakBuffer[100] = { 0 }; // 음성으로 출력할 데이터 저장을 위한
52  int inputBufferIndex = 0;
53  int speakBufferIndex = 0;
54  int inputArr[6] = { 0 }; // 점자 입력 6개 저장을 위한 Array
55  int inputIndex = 0; // inputArr의 마지막 데이터 위치를 저장하는 변수
56
57  //int BUSY = 5;    // JP1-5
58  int PLAY = 6;    // JP1-6
59  int NEXT = 7;    // JP1-7
60  int RESET = 2;    // JP2-5
61
62  // input의 main 함수
63  void InputTask(void) { ... }
64
65  // inputArr에 점자입력을 저장하는 함수
66  void Input(int p1, int p2, int p3) { ... }
67
68  // InputTask에서 6개의 입력을 2진수에서 10진수로 바꿔주는 함수
69  void input_BinToDec(void) { ... }
70
71  // 초성, 모음, 종성을 확인하는 함수
72  int detectData(int a) { ... }
73
74  // 데이터를 합쳐서 글자로 만드는 함수 (파라미터 1개)
75  int combineData(int a) { ... }
76
77  // 데이터를 합쳐서 글자로 만드는 함수 (파라미터 2개)
78  int combineData(int a, int b) { ... }
79
80  // 데이터를 합쳐서 글자로 만드는 함수 (파라미터 3개)
81  int combineData(int a, int b, int c) { ... }
82
83  // Read의 main 함수, inputtask를 통해 받아들이는 데이터를 글자에 해당하는 숫자로 저장
84  void ReadTask() { ... }
85
86  // Speak의 main 함수, Readtask를 통해 저장된 데이터를 해당하는 음원으로 출력
87  void SpeakTask() { ... }
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

4) 설계 작품 완성본

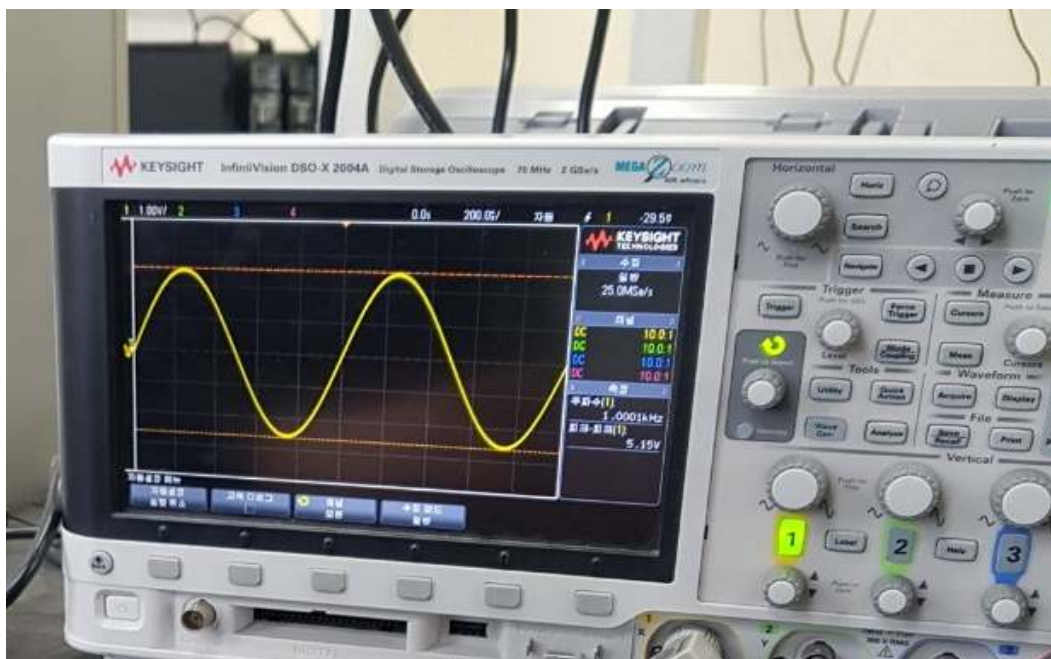


9. 설계작품의 시험 및 성능 결과

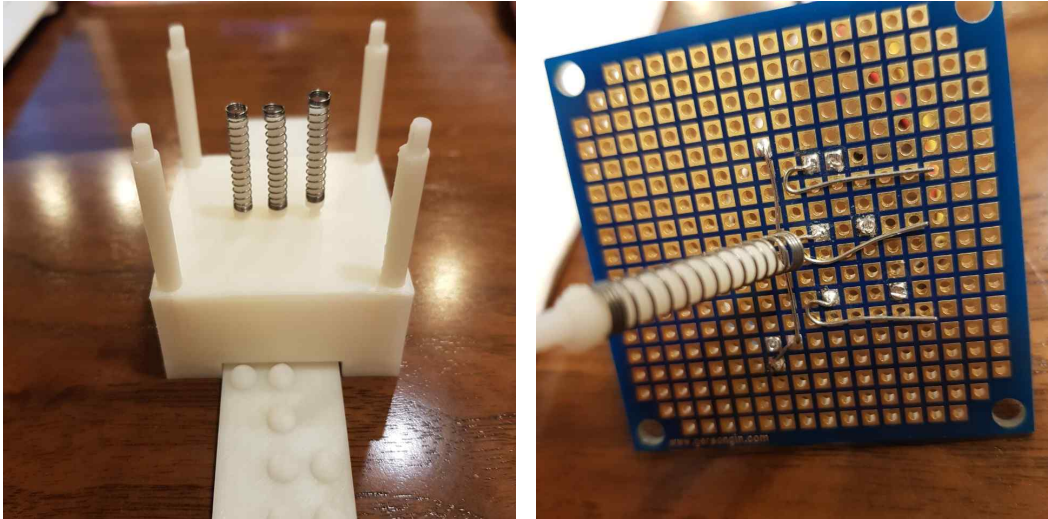
1) 성능

a. 정확한 신호 전달

Short시 신호가 잘 전달되는 지 함수 발생기와 오실로스코프를 통해 확인함



함수 발생기로 5Vpp의 정현파를 발생시키고 점퍼 선에 연결했을 때
신호가 제대로 전달되어 위와 같이 오실로스코프에 출력되는 것을 확인
했다.



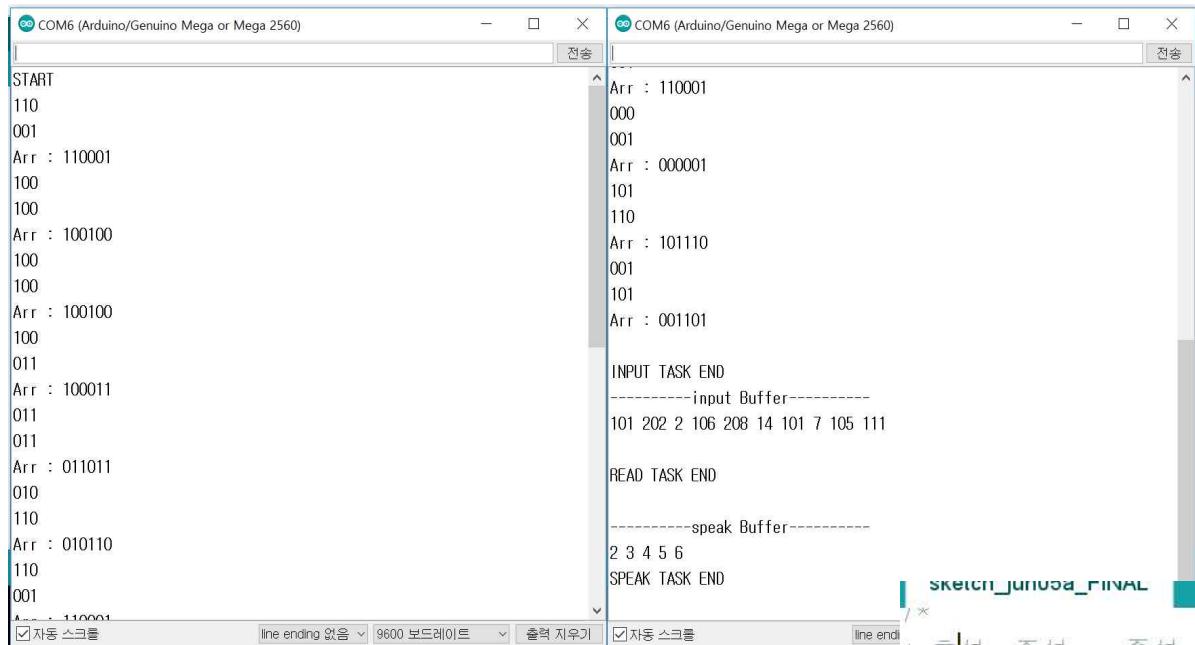
이를 통해 스프링이 상승하여 PCB 기판에 접촉했을 때 Short 되어 신호가 넘어가면 1로 받아들이고, 그렇지 않다면 0으로 받아들여 각 글자에 해당하는 여섯 자리 이진수를 얻어낼 수 있다.

b. 글자를 하나의 단어로 조합

여섯 자리의 이진수를 코딩을 통해 십진수로 변환하여 글자를 읽어내고 십진수 값을 통해 어떤 단어를 입력 받았는지 알 수 있다.

c. 단어에 맞는 음성 파일을 스피커로 출력

한 문장의 인식이 종료되면 앞에서부터 한 단어씩 음성으로 출력한다. 사전에 한 글자씩 녹음한 음원 파일을 오디오 모듈에 저장해놓고, 인식된 글자에 해당하는 음원 파일을 검색하여 재생한다.



위 사진은 "안녕하세요"의 점자를 점자 인식기로 인식하여 스피커에서 "안녕하세요"가 나올 때 Serial 통신을 통해 입력과 저장이 잘 되었는지 확인한 것을 보여준다.

오른쪽 사진은 초성, 중성, 종성에 따른 한글을 숫자에 대입한 것으로, 코딩을 통해 배열에 저장하였다. 위 사진의 input buffer에 저장된 숫자를 보면 "안녕하세요"가 저장된 것을 알 수 있다. "아, 요" 같은 경우는 초성에 "ㅇ"이 들어가서 "ㅇ" 초성 없이 바로 모음이 나온다.

* 초성	중성	종성
* 1	201	101
* 2	202	102
* 3	203	103
* 4	204	104
* 5	205	105
* 6	206	106
* 7	207	107
* 8	208	108
* 9	209	109
* 10	210	110
* 11	211	111
* 12	212	112
* 13	213	113
* 14	214	114
*		115
*		116
*		117

Speak Buffer의 경우 음원 "안", "녕", "하", "세", "요"가 각각 2번째 3번째 4번째 5번째 6번째 파일로 저장되어있어서 2,3,4,5,6이 저장되었다.

"안녕하세요"의 점자를 점자 인식기로 sliding하여 음성 출력을 시작하는

데까지 걸리는 시간은 1초 이내로 매우 빠르나, 오디오 모듈의 한계로 인해 한 글자를 출력하는데 약 1초정도의 시간이 소요되어 하나의 문장에 대해 생각만큼 빠른 속도의 출력은 얻지 못했다. 그러나 점자가 나타내는 단어를 정확하게 출력하는 것은 가능하다.

2) 비용

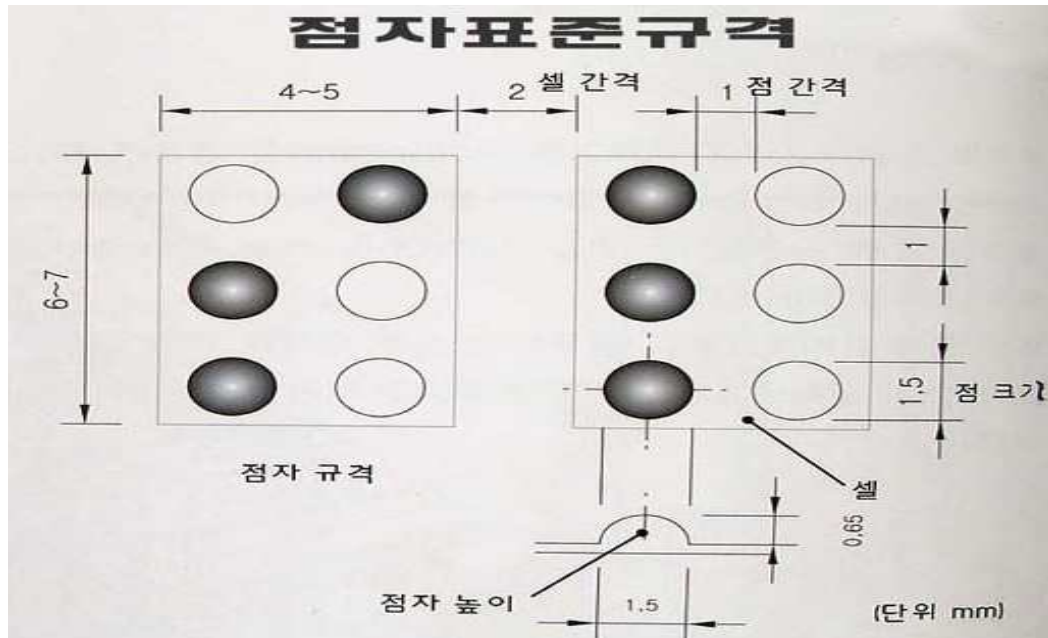
- 3D 프린팅을 통해 인식기를 제작함으로써 비용 부담 최소화
- 아두이노, 오디오 모듈, 스피커 모듈 등 사용된 부품의 금액은 약 10만원 정도로 제한요구사항을 만족함.

3) 환경

- 일정한 속도로 인식기를 sliding 하게 되면 점자가 있는 경우, 하나의 열이 모두 비어있는 경우, 점자 입력이 종료되는 시점 등을 구분해 낼 수 있다.
- 온도와 습도를 다르게 설정하여 동작시켜 볼 수 없어서 이는 제대로 확인하지 못했다.

4) 디자인

a. 점자

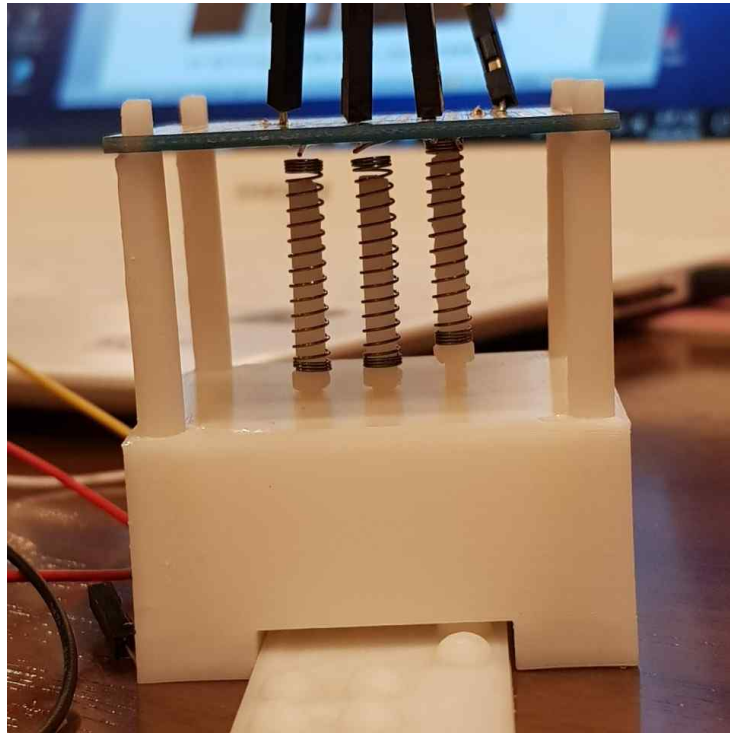


실제 점자 규격은 위와 같다. 하지만 위의 규격으로 점자를 디자인하면 너무 작기 때문에 이에 맞는 부품을 구하기 어렵고 인식률도 많이 떨어질 것을 고려하여 좀 더 큰 사이즈로 디자인 했다.

인식기 모델 사이즈에 맞추기 위해 하나의 점자는 길이 6mm, 높이 2mm로 디자인하고 6개의 점자를 포함하는 셀의 규격은 가로 20mm, 세로 25mm로 디자인했다. 실제보다 큰 점자 규격으로 인해 점자를 좀 더 정확하게 인식하는 것이 가능해졌다.

b. 인식기

- 가로, 세로, 높이가 5cm 내외인 직육면체 형태로 디자인하여 손으로 잡고 사용하기 편하다.



- PCB 기판을 스프링이 올라오는 높이에 맞게 올려놓기 위해 지지대를 3D 프린팅을 통해 정교하게 제작했고, 점자를 지나갈 때 스프링이 정확하게 접촉하여 신호가 전달되는 것을 확인함.
- 인식기 밑 부분에 점자를 지나갈 수 있는 길을 형성하여 인식기가 점자에 걸리거나 이탈하지 않고 부드럽게 지나가는 것을 확인함.

10. 8항의 시험 결과가 4항의 제한요구사항을 만족하는가?

평가 항목	목표	가중치 (%)	평가 기준안
성능	<p>입력 받은 여섯 자리 이진수를 십진수로 변환하여 정확한 글자를 찾도록 설계</p> <p>읽어낸 글자를 하나의 단어로 조합하고 오디오 모듈에서 해당하는 음성 파일을 재생시켜 스피커로 출력</p> <p>데이터를 입력 받은 후 5초 이내에 음성 출력</p> <p>Error rate 3% 미만</p>	40	네 가지 모두 만족하지만 오디오 모듈의 한계로 인해 신속한 음성 출력이 어려움
비용	부품의 금액이 20만원 미만	20	만족
환경	<p>일정한 속도로 sliding 하는 경우 빈 곳을 정확하게 구분</p> <p>습도나 온도에 큰 영향을 받지 않도록 설계</p>	10	습도와 온도의 영향 확인 불가
디자인	<p>점자는 실제 규격보다 크게 디자인</p> <p>사람 손 크기에 맞는 인식기 제작</p> <p>인식기에 점자가 지나가는 길을 형성</p> <p>PCB 기판에 스프링이 정확히 접촉될 수 있도록 설계</p>	30	네 가지 모두 만족

11. 업무 분장

	김남혁	백지현	진장혁
프로젝트 설계	Secondary	Secondary	Primary
시스템설계규격 작성	Secondary	Primary	Secondary
모듈설계규격 작성	Secondary	Secondary	Primary
점자 및 인식기 설계	Primary	Secondary	Secondary
3D 프린팅	Primary	Secondary	Secondary
점자 인식 알고리즘	Secondary	Primary	Secondary
소프트웨어 최적화	Secondary	Primary	Secondary
계획서 / 보고서 작성	Primary	Secondary	Secondary
최종 발표	Secondary	Secondary	Primary

12. 설계 진행일정

구 분	추진 일정									진도율 (%)
	주	2	4	6	8	10	12	14	16	포트폴리오
설계 세부 업무										
자료수집 기술분석 설계 문제 정의		←→								100
설계계획서 작성 및 제출		←→								100
시스템설계규격 작성			←→							100
모듈(인식기) 설계규격 작성				←→						100
H/W 및 S/W 상세설계				←→						100
중간보고서 제출 발표				←→						100
모듈 시험 시스템 시험 성능시험 결과 제한조건요구사항 확인시험					←→					100
최종 보고서 발표							←→			100
학습성과 포트폴리오 업로드 공학인증 사정보고서 제출					←→					100
총 진도율		30			40			30		100